

Practical Assignment 0

Name - Romit Mondal

Roll No. - 18075051

Branch - CSE (B.tech)

Aim: Write a program with a nice UI to Encipher / Decipher with the simple encryption algorithm discussed in the class.

Algorithm Used: Used Atbash cipher which is a substitution cipher with just one specific key where all the letters are reversed that is A to Z and Z to A.

Tech - Stack Used:

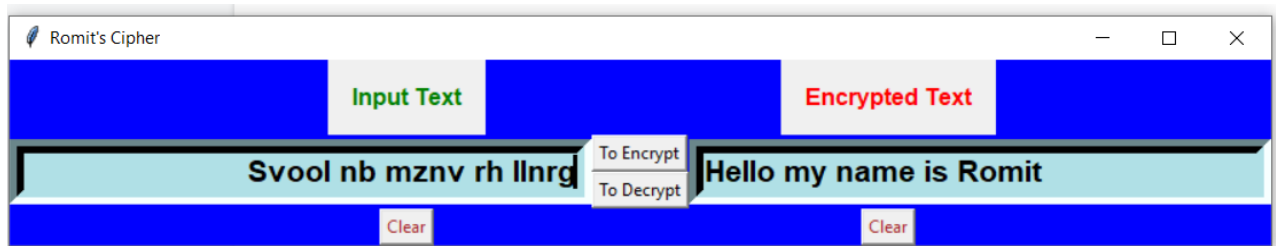
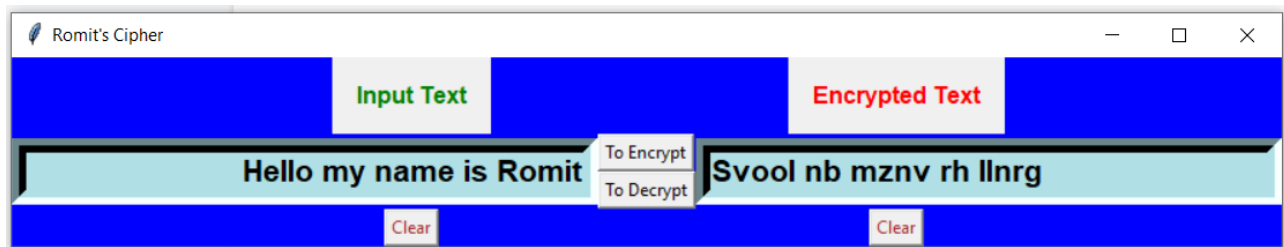
Language - Python 3

OS - Windows 10

Library - Tkinter

IDE - Sublime text

UI Screenshots:



Github Link: https://github.com/Romit10/NetworkSecurity_PA0.git

Code in Python:

```
def Romitencipher(plaintext):
    ciphertext = ""
    for c in range(len(plaintext)):
        char = plaintext[c]
        if (char.isupper()):
            num = 90-(ord(char)-65) % 26
            ch = chr(num)
            ciphertext += ch
        elif char==" ":
            ciphertext += " "
        elif (char.islower()):
            num = 122-(ord(char)-97)%26
            ch = chr(num)
            ciphertext += ch
        else:
            ciphertext += char
    return ciphertext
```

```
def Romitdecipher(ciphertext):
    plaintext = ""
    for i in range(len(ciphertext)):
        char = ciphertext[i]
        if (char.isupper()):
            num = 90-(ord(char)-65) % 26
            ch = chr(num)
            plaintext += ch
        elif char==" ":
            plaintext += " "
        elif (char.islower()):
            num = 122-(ord(char)-97)%26
            ch = chr(num)
            plaintext += ch
        else:
            plaintext += char
    return plaintext
```

```
import tkinter as tk
from tkinter import ttk, Button, Entry
```

```
class RomitCipher:

    def __init__(self, root):
```

```

self.plain_text = tk.StringVar(root, value="")
self.cipher_text = tk.StringVar(root, value="")
self.key = tk.IntVar(root)

root.title("Romit's Cipher")

root.resizable(True, True)

root.configure(background='blue')

style = ttk.Style()
style.configure("TLabel", font = "Serif 20", padding=20)
style.configure("TButton", font="Serif 10", padding=5)
style.configure("TEntry", font="Serif 36", padding=20)

self.plain_label = tk.Label(root, text="Input Text", fg="green", font = ('arial', 12, 'bold'),
                           bd = 16, anchor = "w").grid(row=1, column=1)

self.plain_entry = tk.Entry(root, font = ('arial', 16, 'bold'), textvariable = 'Msg', bd = 10, insertwidth
= 4, bg = "powder blue", justify = 'right', width=32)
self.plain_entry.grid(row=2, column=0, rowspan=2, columnspan=2)
self.plain_clear = tk.Button(root, text="Clear", fg="brown",
                             command=lambda: self.clear('plain')).grid(row=4, column=1)

self.encypher_button = Button(root, text="To Encrypt",
                              command=lambda: self.encypher_press()).grid(row=2, column=3)
self.decipher_button = Button(root, text="To Decrypt",
                              command=lambda: self.decipher_press()).grid(row=3, column=3)

self.cipher_label = tk.Label(root, text="Encrypted Text", fg="red", font = ('arial', 12, 'bold'),
                             bd = 16, anchor = "w").grid(row=1, column=4)

self.cipher_entry = Entry(root,
                          font = ('arial', 16, 'bold'),
                          textvariable = "Result", bd = 10, insertwidth = 4,
                          bg = "powder blue", justify = 'left', width=32)
self.cipher_entry.grid(row=2, column=4, rowspan=2, columnspan=2)

self.cipher_clear = tk.Button(root, text="Clear", fg="brown",
                              command=lambda: self.clear('cipher')).grid(row=4, column=4)

```

```
def clear(self, str_val):
    if str_val == 'cipher':
        self.cipher_entry.delete(0, 'end')
    else:
        self.plain_entry.delete(0, 'end')

def encipher_press(self):
    cipher_text = Romitencipher(self.plain_entry.get())
    self.cipher_entry.delete(0, "end")
    self.cipher_entry.insert(0, cipher_text)

def decipher_press(self):
    plain_text = Romitdecipher(self.cipher_entry.get())
    self.plain_entry.delete(0, "end")
    self.plain_entry.insert(0, plain_text)

root = tk.Tk()
Mohan = RomitCipher(root)
root.mainloop()
```